DOCUMENT RESUME

ED 152 286 IR 005 766

AUTHOR Montgomerie, T. C.; Romaniuk, E. W.
TITLE Interactive Programs: A Simple Way to Allow Naive Users to Manipulate Extensive Data Bases.
INSTITUTION Alberta Univ., Edmonton. Div. of Educational Research Services.
REPORT NO RIR-76-1
PUB DATE 7 May 76
NOTE 24p.; Paper presented to Computer Services Users Applications Symposium (Edmonton, Alberta, May 7, 1976)

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; *Computer Programs; Data Processing; Instructional Aids; *Instructional Systems; *Interaction; *On Line Systems
IDENTIFIERS *IBM 1500 Instructional System

ABSTRACT

This paper describes the IBM 1500 Instructional System which is used for computer assisted instruction (CAI), research, and demonstration at the University of Alberta. The system automatically stores a performance record of each student's responses to instructional programs, and a number of CAI support programs are accessible to perform various analyses of daily data sets for each course. A set of interactive programs which provide assistance to CAI authors make data base manipulations easier: they explain the use of specific programs, query the user as to his requirements, check user input for errors, prompt the user as to legitimate answers, generate a file with the batch job, and execute a call to run the job. Use of the programs has reduced time to construct the file to run programs, reduced sessions to achieve a successful run, and reduced time spent in helping CAI authors. Sample terminal sessions and batch jobs are appended. (KP)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED152286

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

RIR-76-1

INTERACTIVE PROGRAMS:

A Simple Way to Allow Naive Users to Manipulate Extensive Data Bases.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

E.W. Romaniuk

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."

Presented to: Computer Services Users
Applications Symposium
The University of Alberta
May 7, 1976

T.C. Montgomerie
E.W. Romaniuk

Division of Educational Research Services
Faculty of Education
The University of Alberta
Edmonton, Alberta T6G 2G5

IR005-766

INTRODUCTION

The Division of Educational Research Services (DERS) has operated an instructional computing system since 1968. A service division in the Faculty of Education, DERS has as its primary mandate the facilitation of educational research. This is accomplished chiefly by using computers to assist staff members and graduate students in the collection and analyses of research data.

The IBM 1500 Instructional System was originally obtained to provide the most current facilities for research into the improvement of instruction. One of only about twenty such systems produced by IBM for research in and development of computer-assisted instruction (CAI), the IBM 1500 System at DERS is being used for purposes of instruction, research and demonstration.

Essentially an IBM 1130-based system, the 1500 System consists of the following components:

Central Units:

1131 CPU-3.6 ms cycle time with 64k bytes of core storage.

1133 Multiplex Control.

1502 Station Controller.

Storage Devices:

114/115 Logicon/Intercomp Disk Units - two drives with a total of 50 million bytes of on-line storage.

2415 Magnetic Tape Unit - two drives.

I/O Devices:

1132 printer - 88 lines per minute.

1442 card reader/punch - 80 cards per minute.

Instructional Stations:

20 stations each consisting of:

-CRT (16 lines x 40 characters, half:line feature, access to 512 programmable characters)

-Light Pen

-Image Projector (1024 black and white or color addressed still frames on 16mm film.

-Audio Play/Record Unit (two hours of pre-recorded audio on addressed audio tape)

2 stations consisting of a typewriter terminal

Interactive Languages:

Coursewriter II and APL/1500.

STUDENT PERFORMANCE RECORDS

One of the important features of the 1500 System, from an instructional point of view, is that each time a student responds by typing an answer or pointing with the light pen, the system can automatically save a 360 byte "student performance" record on magnetic tape. A performance record contains information such as the student's typed response (or CRT coordinates of the area pointed at), question

identifier, time taken to respond, date and time, student's name, response category matched (i.e., correct, wrong, additional, unrecognizable) and the contents of switches and counters. On the average a student will enter approximately two responses per minute or 120 responses per hour. Assuming fifteen terminals are in use ten hours per day, the total number of 360 byte performance records written is approximately 18,000 per day. Based upon 22,000 student contact hours of instruction in 1975 and 120 responses per hour per student, it was possible to have saved four million performance records. At this point in time, a number of the available courses do not save many performance records and this helps to minimize the number of records and data sets to be maintained.

At the end of each day, in batch mode on the 1130 system, the tape containing the day's accumulated performance records is divided into as many data sets as CAI courses accessed by students that day. These data sets are then written to a magnetic tape (the *1500* tape) and sent to the Department of Computing Services. Because of the many analyses required, a number of CAI Support (CAIS) programs have been written in FORTRAN and PL/1. These support programs, contained in the DERS CAIS library (see Figure 1), vary in complexity from a simple program to reformat a magnetic tape containing a CAI course listing, to a relatively complex set of programs to provide summary Performance Analyses. The relationship between some of

these CAIS programs and the magnetic tapes and files used and produced by them is shown in Figure 2.

CRT DISPLAYS	CWBUILDER
DOCUMENTER	FLOWCHARTER
IDCHANGER	LISTER
PERFORMDUMP	PERFORMGROUP
PERFORMINDIV	PERFORMPLOT
PERFORMSUM	TOTALPRTAPE
TAPEUPDATE	

Figure 1. Programs in the CAIS Interactive Library.

THE CAIS INTERACTIVE LIBRARY

Most CAI course authors are not sophisticated computer users. As these programs make extensive use of user specified parameters and require that input be in a specific form, probably requiring that first at least one *SORT routine be run on the input file, it was found that staff at DERS spent much of their time assisting authors in running the programs. Even then, due to the complexity of some of the programs, it was found that costly errors (both financial and time) were being made. These were primarily due to errors in parameter input or simply due to non-optimal procedures.

In order to solve these problems, a set of interactive

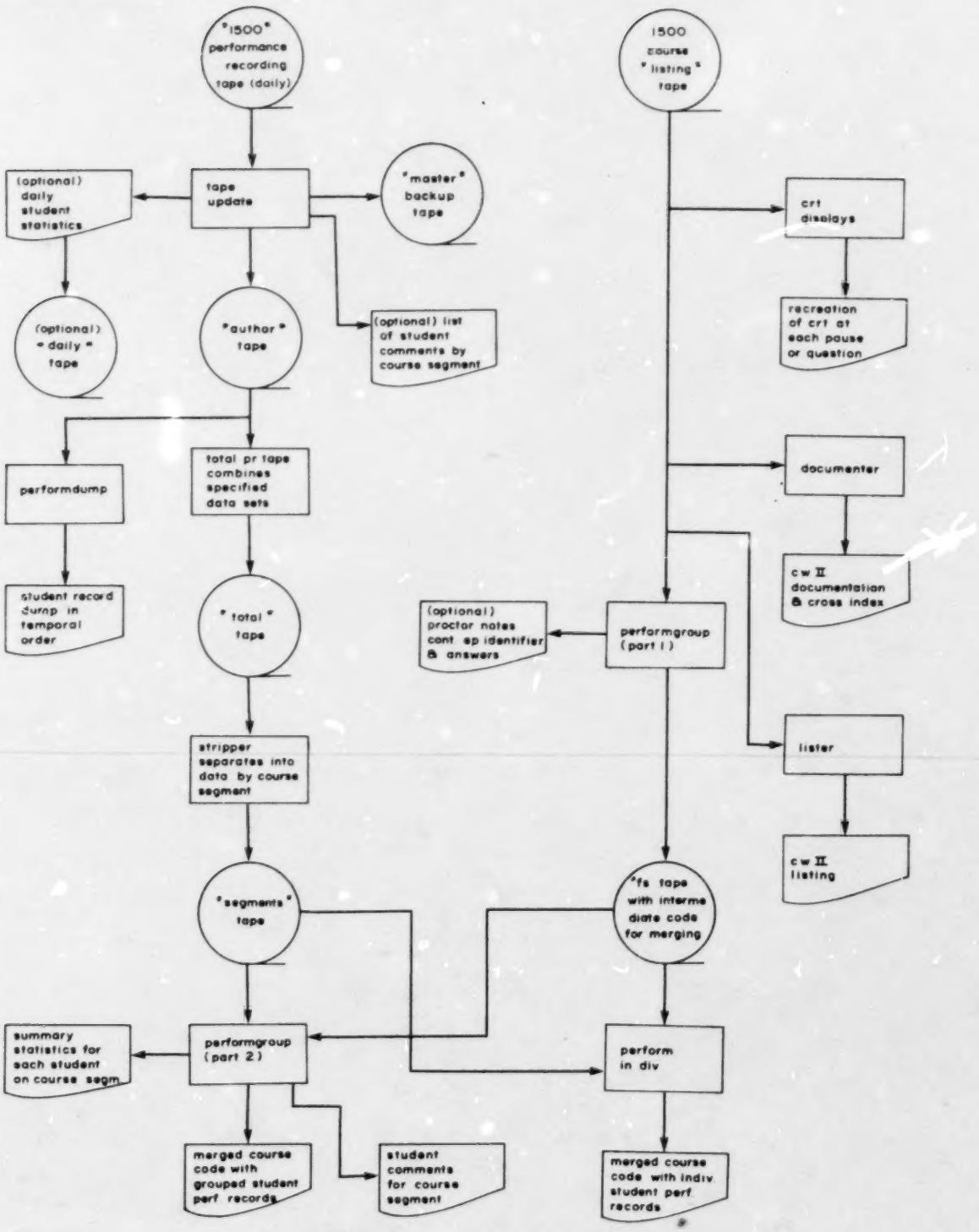
"1500" performance recording tape (daily)
1500 course "listing" tape
(optional) daily student statistics
tape update
"master" backup tape
crt displays
(optional) "daily" tape
"author" tape
(optional) list of student comments by course segment
recreation of crt at each pause or question
performdump
total pr tape combines specified data sets
documenter
student record dump in temporal order
"total" tape
(optional) proctor notes cont ap identifier & answers
performgroup (part 1)
cw II documentation & cross index
lister
stripper separates into data by course segment
cw II listing
"segments" tape
"fs tape with interme diate code for merging
summary statistics for each student on course segm.
performgroup (part 2)
perform in div
merged course code with grouped student perf records
student comments for course segment
merged course code with indiv. student perf records

Figure 2. Flow Diagram of DERS CAI Author Support Programs.

programs were written. These programs were designed to:

-explain the use of specific programs,

-query the user as to his requirements,

-check user input for errors,

-prompt the user as to legimate answers,

-estimate the requirements for and generate the $SIGNON card,

-generate a file with the *BATCH job in it, and

-execute a call to *BATCH to run the job.

Due to the fact that the MTS command language is automatically generated, it was decided that many of the '$RUN' instructions should have a '$IF' ... '$GOTO' sequence associated with it. This feature is extremely effective in saving dollar costs where errors (either operating or programming) are encountered.

An example of an interaction in which an author desires to run the grouped student analysis program (PERFORMGROUP) is shown in Appendix I. The MTS *BATCH job which would be generated by this interactive session is shown in Appendix II. As can be seen from Appendix I, the input required from the users is kept as simple as possible. Wherever possible all user input is checked for validity, and the user is asked to correct obvious errors. At the end of the interactive session, the user is asked about the status of any temporary tapes he may have used. Depending on the user response, a line may be written in a file in the 1500 Operator's CSID, notifying the operators to recall and re-

use a tape.

As an example of the effectiveness of the interactive program, a user wishing a group analysis of a single set of student performance data will spend approximately five minutes at a terminal. The cost of a typical interactive session is about sixty cents. The resulting MTS command file of approximately 150 lines (See Appendix II) will contain instructions to:

-run nine mainline programs,

-issue six calls to *SORT,

-mount a total five tapes,

-check most $RUN commands using $IF ... $GOTO, and

-issue tape recall and/or *MOUNT error messages.

CAIS: TAPEUPDATE

Prior to any analyses of student performance data, the DEPS CAI operators must first utilize CAIS:TAPEUPDATE. This step is necessary in order to copy the *1500* tape to two different tapes; the first tape is the *MASTER* tape which is a cumulative copy of all the daily data sets. The second tape is called the *AUTHOR* tape an is unique to each course. When writing to these tapes, each data set is appended as the last sequential data set on that tape. In this way, the records of a semester course are maintained in daily data sets on a single *AUTHOR* tape. Authors who wish to analyze any particular data set(s), always have access to

their *AUTHOR* tapes.

Most authors generally request three different kinds of analyses. The first analyses is an option in the program CAIS:TAPEUPDATE. It provides daily status information about the students registered in the course. Information returned to the author includes the id's of the students who accessed the course that day, the length of time each student was signed-on to the course, the number of correct responses entered by each student and the current location of each student in the course. In addition, a sorted list of all student initiated comments made while on-line that day will be provided for the author. The author also has the option of writing certain portions of the performance records to the *DAILY* tape. Occasionally the students will ask the author questions related to the previous CAI session. In order to trace the student's progress for that session, the author only needs to sit at a terminal with access to MTS, mount *DAILY*, and scan the file sorted by student-by-time.

CAIS: PERFORMDUMP

The second set of analyses which can be performed on the student performance records are available through the CAIS interactive program PERFORMDUMP. This program was designed essentially for use by DERS CAI Consultants and System Analysts. PERFORMDUMP sorts the daily student performance data set (found on the *AUTHOR* tape) into one

of two orders:

1) ascending order by time.

2) ascending order by student id by time.

Following the sort, each performance record is then decoded, formatted, and printed. This provides an absolute trace of the students on a particular day. While the output could be used by an author to analyse a student's progress through a course, it has been found to be too inconvenient to use for this purpose. Instead, the output from PERFORMDUMP is used to monitor information pertaining to a student or a group of students at the time an unexpected "happening" occurred. This information is often necessary when trying to uncover elusive programming or systems errors.

CAIS: PERFORMGROUP

The program PERFORMGROUP provides a very useful tool for the CAI author. The effect of the program is to merge the information contained on the *AUTHOR* tape with a *LISTING* tape. The resulting output shows the contents of the CRT each time a question was asked, and shows in tabular form, the number of students choosing each of the available answers.

Program PERFORMGROUP consists of two parts which are run through terminal access to MTS. Each part can be run in

separate interactive sessions or continuously in one session. The two parts are:

1) An analysis of the CWII course listing (*LISTING* tape) to set up the proper format for merging with the performance records on the *AUTHOR* tape.

2) A merging of the performance tape (*AUTHOR*) with the data set of intermediate code prepared in Part 1.

Three options are available with the program. First, the actual student response for each response type (correct answer, unanticipated answer, etc.) may be listed.

The second option allows the analyses of all student responses, or only the responses a student gave on a specific pass through a question. For example, specification of PASS 1 would include each student's responses up to and including the first correct response for each question. This information allows the author to see how adequately his answer analyses cover the range of student responses. If improvements to the answer analyses are required, these can be done on the basis of student performance data.

The third option generates a set of "Proctor Notes." The "Proctor Notes" include a list of the epid's and the answers associated with each epid. This information enables a Proctor to assist students during times when the course author is not available.

Two further tables are output with Part 2 of

PERFORMGROUP. They are:

1) A listing, sorted by epid, of all student comments, and

2) A summary of student statistics for the course segment.

Once Part 1 has been completed, the resulting intermediate code is automatically saved on a *FS tape. Part 1 needs to be re-run only when the course has been modified. The intermediate code may be used as many times as necessary in running Part 2. Because Part 1 is by far the most expensive part of the total program, this two-part separation of the program can save a considerable amount of money.

SUMMARY

There is no doubt that the use of the interactive CAIS programs has greatly aided DERS CAI authors. The time taken to construct the *BATCH file (varying from 30 to 400 or more lines) to run the programs is often less than five minutes. This is a vast reduction from the time previously required to construct and modify "model run" files. In addition the interactive programs help keep down the accumulation of files to be saved until the job has run successfully. The number of different sessions required until a successful run has been achieved is now most often only one. If the author provides the correct parameters, as required by the interactive programs, chances are that the job will run

successfully on the first trial.

Finally, the interactive programs have released the DEPS Consultants from the time consuming job of trying to help CAI authors with MTS, *MOUNT parameters, *SORT parameters, various CAIS programs, etc. If any errors have occured in the *BATCH run, the assistants can quickly query the various data sets because the data set names and record formats have been standardized.

Without access to a sophisticated terminal system such as MTS, it would have been difficult to compile the available interactive CAIS programs. Those user groups providing service to naive users, especially where numerous data base files are to be manipulated, should certainly look to the possibility of doing so through interactive programs.

APPENDIX I
SAMPLE TERMINAL SESSION

OPTIONS AND ERROR CHECKING IN PERFORMGROUP

THE CAIS INTERACTIVE LIBRARY

Would you like to see the available programs? YES

The CAIS library consists of a number of separate programs designed to aid the CAI author. These programs may be run under *BATCH mode on MTS using this interactive system.

The programs presently accessible from this system are:

CRTDISPLAYS	CWBUILDER	DOCUMENTER	FLOWCHARTER
IDCHANGER	LISTER	PERFORMDUMP	PERFORMGROUP
PERFORMINDIV	PERFORMPLOT	PERFORMSUM	TOTALPRTAPE
TAPEUPDATE			

Do you wish to have any of the CAIS programs explained? NO

I cannot recognize your response, please type YES OR NO: NO

Do you wish to run any of the CAIS programs? YES

What is the name of the program you wish run? PERFORMGROUP

PROGRAM PERFORMGROUP

Do you understand the difference between Part 1 and Part 2 of Program PERFORMGROUP? YES

Do you want to run Part 1 or Part 2. (Just type 1 or 2.) 1

Do you have the CWII COURSE LISTING TAPE FORM which was returned with your 1500 Service Request Form? YES

Please enter your Sign-On ID: CSID

Please enter your password: ******

Do you wish this output returned to Education? YES

What is your Bin number? 22

How many copies of the output do you wish? 3

What priority do you wish? D

On the CWII COURSE LISTING TAPE FORM is a "VOLUME SERIAL" beginning with ER.

APPENDIX I
SAMPLE TERMINAL SESSION

Please fill in the four digits - ER 5021

I don't think you gave me the right number.
It begins with a 6 and there are no blanks. Try again ER 6021

Is there more than one file on the tape? YES

In what position on the tape is this file? 3__

What is the 5 character name of your course? MONEY

What is the three digit segment number? (Please put in all leading zeros, eg. 000, OR 020). 003

Would you like a set of ep's and expected answers for the Proctor? YES

Do you wish all correct responses printed? NO_

Do you wish all wrong responses printed? NO_

Do you wish all additional responses printed? NO_

Do you wish all unanticipated responses printed? YES

Do you wish all overtime responses printed? YES

Do you wish lightpen responses printed? NO_

You may have all student responses tabulated, or only the responses for the first pass of each student through each section. Do you want ALL student responses tabulated? YES

Do you wish to use any special character for overstriking alternate coded characters? (The default is a "bullet".) YES

What character do you want to use? $

***** THIS FILE HAS BEEN CALLED *****
PRCODE.003.MONEY
***** PLEASE MAKE A NOTE OF THE NUMBER *****

Since there is more than one file on this tape, do you wish to run PERFORMGROUP on another file? YES

In what position on the tape is the next file? 77_

What is the 5 character name of your course? MONEY

What is the three digit segment number? (Please put in all leading zeros, eg. 000, OR 020). 000

Would you like a set of ep's and expected answers for the Proctor? YES

APPENDIX I
SAMPLE TERMINAL SESSION

Do you wish the responses printed to be the same as for the previous file? YES

***** THIS FILE HAS BEEN CALLED *****
PRCODE.000.MONEY
***** PLEASE MAKE A NOTE OF THE NUMBER *****

Do you wish to run PERFORMGROUP on any more files? YES

In what position on the tape is the next file? 6

What is the 5 character name of your course? STAT1

What is the three digit segment number? (Please put in all leading zeros, eg. 000, OR 020). 019

Would you like a set of ep's and expected answers for the Proctor? YES

Do you wish the responses printed to be the same as for the previous file? NO

Do you wish all correct responses printed? NO

Do you wish all additional responses printed? YES

Do you wish all unanticipated responses printed? YES

Do you wish all overtime responses printed? YES

Do you wish lightpen responses printed? YES

You may have all student responses tabulated, or only the responses for the first pass of each student through each section. Do you want ALL student responses tabulated? YES

***** THIS FILE HAS BEEN CALLED *****
PRCODE.019.STAT1
***** PLEASE MAKE A NOTE OF THE NUMBER *****

Do you wish to run PERFORMGROUP on any more files? NO

Are you ready to run Part 2 now? YES

Before you run Part 2 on any set of performance data, you MUST have had that data separated into segments. Has this been done yet? NO

Before you go any further, you must arrange for a tape which will contain your performance sorted into individual segments. this will be referred to as your "SEGMENTS" tape. Have you arranged for this tape yet? NO

APPENDIX I
SAMPLE TERMINAL SESSION

You cannot run Part 2 until this tape has been obtained.
At this point you can:
1) Cancel this whole job.
2) Arrange for the tape immediately - I will wait for you. (Don't "Interrupt" or "SIGN OFF" - just go get the tape.)
3) Just run Part 1 (If it has been specified previously).
Type the number of your choice. 2

OK - Go get the tape. Hit "RETURN" when you return.

Did you get the tape? YES

Let's look first at the "AUTHOR" tape
What is the rack number? E0 6620
Now the "VOLUME SERIAL" ER 1234

Four options for merging files are available:
1) All files on the input tape may be merged.
2) Selected files on the input tape may be merged.
3) All files on the input tape, beginning at a specified file, may be merged.
4) Files "M" to "N" inclusive on the input tape may be merged.
Type in the number of the option you wish. 3

What is the position of the first file you wish to have merged? 3

Now I need to know about your "SEGMENTS" tape.
What is the Computing Services Rack Number? E0 6259

What is the "Volume Serial"? ER 0609

What is the five character name of the first course for which Performance Records are stored on this tape? STAT1

Now type in ALL the segments of this course for which Performance Records exist. Type the segments in one at a time. All leading zeros must be typed (eg. 021).
type END when all segments have been entered.
Enter a segment 003
Enter a segment 019
Enter a segment END

Do you wish to recover the Performance Records for any other courses from this tape? YES

Enter another course name. NEWCR

Now type in ALL the segments of this course for which Performance Records exist. Type the segments in one at a time. All leading zeros must be typed (eg. 021).
type END when all segments have been entered.
Enter a segment 000
Enter a segment END

APPENDIX I
SAMPLE TERMINAL SESSION

Do you wish to recover the Performance Records for any other courses from this tape? NO_

Do you want to run Part 2 against all the segments you ran Part 1 against today? NO_

Please type the "PRCODE" of a file on which you wish to run Part 2.
PRCODE.999.STAT1

That is not a legitimate code which was given to you today. Please try again.
PRCODE.019.STAT1

Do you wish to run Part 2 on any other segments which were generated during this run? NO_

Do you wish to run Part 2 against any segments which have been run on Part 1 previously? YES

Please type in the "PRCODE" of the first segment you wish to generate performance data for. PRCODE.003.STAT1

You may have all student responses tabulated, or only the responses for the first pass of each student through each section. Do you want ALL student responses tabulated? YES

Do you wish to run another segment? NO_

Will you be using the course listing tape for other jobs? NO_

APPENDIX II
SAMPLE *BATCH JOB

SAMPLE OUTPUT

```
$SIGNON CSID P=1000 9TP=3  PRIO=D PRINT=TN T=15M COPIES=3 -
        RETURN=EDUC 'BIN = 22 '
PSWORD
$RUN *MOUNT PAR=E06620 9TP *OUT* VOL=ER1234 QUIT=NO
$IF RC>0 GOTO ERR1
$RUN *MOUNT PAR=E06300 9TP *IN* VOL=ER1618 RING=IN QUIT=NO -
     FMT=FB LP=ON BLKSIZE=7200 LRECL=360
$IF RC>0 GOTO ERR2
$CONTROL *OUT* POSN *3*
$RUN CAIS:TOTALPRTAPE 1=*OUT* 2=*IN*
$RUN *MOUNT PAR=E06259 9TP *OUT* VOL=ER0609 RING=IN
$IF RC>0 $GOTO ERR10
$DESTROY -A
$DESTROY -B
$CREATE -A TYPE=SEQ SIZE=1000P
$CREATE -B TYPE=SEQ SIZE=1000P
$GET -SEG
$NUMBER
003
003
$UNNUMBER
$CONTROL *OUT* FMT FB(7200,360)
$CONTROL *OUT* DSN 003.STAT1
$CONTROL *IN* REW
$RUN CAIS:STRIP.SEG++PL1LIB PAR=SEG_NO=-SEG -
     PERFORM=*IN*@FB(7200,360) OUTPUT=-B@FB(7200,360)
$RUN *SORT PAR=SORT=CH;A;33;10;CH;A;9;17 -
     INPUT=-B;FB;360;7200 OUTPUT=*OUT*;FB;360;360 MNR=5000
$CONTROL *OUT* WTM
$DESTROY -A
$DESTROY -B
$CREATE -A TYPE=SEQ SIZE=1000P
$CREATE -B TYPE=SEQ SIZE=1000P
$GET -SEG
$NUMBER
019
019
$UNNUMBER
$CONTROL *OUT* FMT FB(7200,360)
$CONTROL *OUT* DSN 019.STAT1
$CONTROL *IN* REW
$RUN CAIS:STRIP.SEG++PL1LIB PAR=SEG_NO=-SEG -
     PERFORM=*IN*@FB(7200,360) OUTPUT=-B@FB(7200,360)
$RUN *SORT PAR=SORT=CH;A;33;10;CH;A;9;17 -
     INPUT=-B;FB;360;7200 OUTPUT=*OUT*;FB;360;360 MNR=5000
$CONTROL *OUT* WTM
$DESTROY -A
$DESTROY -B
$CREATE -A TYPE=SEQ SIZE=1000P
$CREATE -B TYPE=SEQ SIZE=1000P
$GET -SEG
$NUMBER
000
```

APPENDIX II
SAMPLE *BATCH JOB

```
000
$UNNUMBER
$CONTROL *OUT* FMT FB(7200,360)
$CONTROL *OUT* DSN 000.NEWCR
$CONTROL *IN* REW
$RUN CAIS:STRIP.SEG+*PL1LIB PAR=SEG_NO=-SEG -
     PERFORM=*IN*@FB(7200,360) OUTPUT=-B@FB(7200,360)
$RUN *SORT PAR=SORT=CH;A;33;10;CH;A;9;17 -
     INPUT=-B;FB;360;7200 OUTPUT=*OUT*;FB;360;360 MNR=5000
$CONTROL *OUT* WTM
$COMMENT   RELEASE THE TAPE - GET RID OF BLOCKING PROBLEM
$RELEASE *IN*
$RUN *MOUNT PAR=E06021 9TP *IN* VOL=ER6021
$IF RC>0 $GOTO ERROR
$CONTROL *IN* POSN *3*
$IF RC>0 $GOTO ERROR
$DESTROY -A
$DESTROY -B
$CREATE -A SIZE=200P
$CREATE -B SIZE=200P
$CREATE -P
$NUMBER
FAKE TITLE CARD
MONEY003           $                    1    1    1    1
$$CONTINUE WITH *IN*
$UNNUMBER
$RUN QZZP:CAISOBJ+QZZP:CWLIB.O+XDER:SUB 5=-P  6=-A -
     7=-B 18=CAIS:USERS(LAST+1)
$IF RC>0 $GOTO ERR
$RUN *MOUNT PAR=E06258 9TP *FS* VOL=ER0201 RING=IN QUIT=NO
$IF RC>0 $GOTO FSERR
$RUN *FS 0=*FS*
SAVE -B PRCODE.003.MONEY
STOP
$CREATE -AA SIZE=500P TYPE=SEQ
$CREATE -BB SIZE=500P TYPE=SEQ
$CREATE -CC SIZE=500P TYPE=SEQ
$CREATE -D SIZE=500P TYPE=SEQ
$CR -E SIZE=500P TYPE=SEQ
$CR -F SIZE=500P TYPE=SEQ
$CREATE -Z SIZE=200P
$CREATE -COMMENT SIZE=100P TYPE=SEQ
$CREATE -SORTCOM SIZE=100P TYPE=SEQ
$EM -EP
$EM -EPSORT
$EM -PRINTIT
$RUN CAIS:EPLIST+*PL1LIB PAR=CARD=-B PRINT=-PRINTIT@F(132) -
    EPS=-EP@F(15) EP_SORT=-EPSORT@F(15)
$COPY -PRINTIT
$CONTROL *IN* POSN *77*
$IF RC>0 $GOTO ERROR
$EMPTY -A
$EMPTY -B
$EMPTY -P
$GET -P
```

APPENDIX II
SAMPLE *BATCH JOB

```
$NUMBER
FAKE TITLE CARD
MONEY000            $                    1    1          1
$$CONTINUE WITH *IN*
$UNNUMBER
$RUN QZZP:CAISOBJ+QZZP:CWLIB.O+XDER:SUB 5=-P  6=-A -
     7=-B 18=CAIS:USERS(LAST+1)
$IF RC>0 $GOTO ERR
$RUN *FS 0=*FS*
SAVE -B PRCODE.000.MONEY
STOP
$EM -EP
$EM -EPSORT
$EM -PRINTIT
$RUN CAIS:EPLIST+*PL1LIB PAR=CARD=-B PRINT=-PRINTIT@F(132) -
     EPS=-EP@F(15) EP_SORT=-EPSORT@F(15)
$COPY -PRINTIT
$CONTROL *IN* POSN *6*
$IF RC>0 $GOTO ERROR
$EMPTY -A
$EMPTY -B
$EMPTY -P
$GET -P
$NUMBER
FAKE TITLE CARD
STAT1019            $    1    1    1    1    1          1
$$CONTINUE WITH *IN*
$UNNUMBER
$RUN QZZP:CAISOBJ+QZZP:CWLIB.O+XDER:SUB 5=-P  6=-A -
     7=-B 18=CAIS:USERS(LAST+1)
$IF RC>0 $GOTO ERR
$RUN *FS 0=*FS*
SAVE -B PRCODE.019.STAT1
STOP
$EM -EP
$EM -EPSORT
$EM -PRINTIT
$RUN CAIS:EPLIST+*PL1LIB PAR=CARD=-B PRINT=-PRINTIT@F(132) -
     EPS=-EP@F(15) EP_SORT=-EPSORT@F(15)
$COPY -PRINTIT
$EMPTY -AA
$EMPTY -BB
$EMPTY -CC
$EMPTY -D
$EMPTY -E
$CONTROL *OUT* POSN 019.STAT1
$EMPTY -Z
$EMPTY -F
$COPY *OUT* -P
$RUN CAIS:GRUPA+*PL1LIB PAR=CARD=-B@F(80) DSKTAB=-AA@F(1958) -
     DSKBUF=-BB@F(113) PERFORM=-F@F(360) DSKTEMP=-E@F(1958)
$RUN *SORT PAR=SORT=CH;A;1;12 INPUT=-BB;F;113;113 -
     OUTPUT=-CC;F;113;113 MNR=5000
$RUN CAIS:GRUPB+*PL1LIB PAR=DSKTAB=-AA@F(1958) -
     DSKBUF=-CC@F(113) DSKPLOT=-D@F(80) PRINT=-Z@F(133)
```

APPENDIX II
SAMPLE *BATCH JOB

```
$COPY -Z
$EMPTY -COMMENT
$EMPTY -Z
$EMPTY -SORTCOM
$RUN CAIS:STRIP.COM+*PL1LIB PAR=PERFORM=-F@F(360) -
     COMMENT=-COMMENT@F(85) PRINT=*DUMMY*
$RUN *SORT PAR=SORT=CH;A;9;25 INPUT=-COMMENT;F;85 -
     OUTPUT=-SORTCOM;F;85 MNR=5000
$RUN CAIS:COMMENTS+*PL1LIB PAR=COMMENT=-D@F(85) -
     SORTREC=-SORTCOM@F(85) PRINT=-Z@F(133)
$COPY -Z
$EMPTY -AA
$EMPTY -Z
$RUN *SORT PAR=SORT=CH;A;9;17 INPUT=-F;F;360;360 -
     OUTPUT=-AA;FB;360;7200 MNR=10000
$RUN CAIS:SEG.SUMMARY+*PL1LIB PAR=PERFORM=-AA@FB(7200,360) -
     CUM_PER=-Z@F(133)
$COPY -Z
$EMPTY -B
$RUN *FS 0=*FS*
RESTORE PRCODE.003.STAT1 -B
STOP
$EDIT -B
0 1 /                                        0/
STOP
$EMPTY -AA
$EMPTY -BB
$EMPTY -CC
$EMPTY -D
$EMPTY -E
$CONTROL *OUT* POSN 003.STAT1
$EMPTY -Z
$EMPTY -F
$COPY *OUT* -F
$RUN CAIS:GRUPA+*PL1LIB PAR=CARD=-B@F(80) DSKTAB=-AA@F(1958) -
     DSKBUF=-BB@F(113) PERFORM=-F@F(360) DSKTEMP=-E@F(1958)
$RUN *SORT PAR=SORT=CH;A;1;12 INPUT=-BB;F;113;113 -
     OUTPUT=-CC;F;113;113 MNR=5000
$RUN CAIS:GRUPB+*PL1LIB PAR=DSKTAB=-AA@F(1958) -
     DSKBUF=-CC@F(113) DSKPLOT=-D@F(80) PRINT=-Z@F(133)
$COPY -Z
$EMPTY -COMMENT
$EMPTY -Z
$EMPTY -SORTCOM
$RUN CAIS:STRIP.COM+*PL1LIB PAR=PERFORM=-F@F(360) -
     COMMENT=-COMMENT@F(85) PRINT=*DUMMY*
$RUN *SORT PAR=SORT=CH;A;9;25 INPUT=-COMMENT;F;85 -
     OUTPUT=-SORTCOM;F;85 MNR=5000
$RUN CAIS:COMMENTS+*PL1LIB PAR=COMMENT=-D@F(85) -
     SORTREC=-SORTCOM@F(85) PRINT=-Z@F(133)
$COPY -Z
$EMPTY -AA
$EMPTY -Z
$RUN *SORT PAR=SORT=CH;A;9;17 INPUT=-F;F;360;360 -
     OUTPUT=-AA;FB;360;7200 MNR=10000
```

APPENDIX II
SAMPLE *BATCH JOB

```
$RUN CAIS:SEG.SUMMARY**PL1LIB PAR=PERFORM=-AA@FB(7200,360) -
     CUM_PER=-Z@F(133)
$COPY -Z
$DESTROY -B
$GET EROP:TAPES
$NUMBER LAST+1
TAPE E06021 MAY BE RECALLED
$UNNUMBER
$SIGNOFF
$ERROR:$GET QZZP:ERRORS
$NUMBER LAST+1
USER CSID HAD A PROBLEM MOUNTING TAPE 6021
$UNNUMBER
$COPY CAIS:TROUBLE.MESS *SINK*
$SIGNOFF
$ERR10:$GET QZZP:ERRORS
$NUMBER LAST+1
USER CSID HAD A PROBLEM MOUNTING TAPE *aaa
$UNNUMBER
$COPY CAIS:TROUBLE.MESS *SINK*
$SIGNOFF
$FSERR:$GET QZZP:ERRORS
$NUMBER LAST+1
USER CSID HAD A PROBLEM MOUNTING TAPE 6258
$UNNUMBER
$COPY CAIS:TROUBLE.MESS *SINK*
$SIGNOFF
$ERR:$LIST -A
$LIST -B
$COPY CAIS:TROUBLE.MESS *SINK*
$SIGNOFF
```